EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for the examiner's amendment was given in a telephone interview with Ms. Elizabeth Ruzich, Registration No. 54,416 on July 23, 2008.

3. Amend the claims as follow:

1. (Currently Amended) A method of inter-process communication between at least two application processes on a single processor of a single computer, comprising the steps of:

a first process of a first application determining a name of a first rendezvous file in a file system of the computer, the name of the first rendezvous file being associated with a second application, the first rendezvous file containing information for the first process to connect to a second process of the second application for inter-process communication;

the first process initiating a first connection to the second process using the information contained in the first rendezvous file;

the first process communicating with the second process using the first connection if the first connection is successfully established to request the second process to perform a task for the first process by transmitting a set of keys and values between said second process and said first process; and the first process starting a third process of the second application if the first process fails to establish a connection with the second process,

wherein multiple instances of an application process combine into one running instance of said application process;

wherein said inter-process communication between at least two application processes is restricted to communications between applications running on a single processor of said single computer under control of a single instance of an operating system.

2. (Currently Amended) The method of Claim 1, further comprising:

the first process initiating a second connection to the third process using the information in the first rendezvous file, in response to the third process informing the first process that the third process is ready for a connection.

3. (Original) The method of Claim 2, wherein the third process is started in a server mode without a user interface.

4. (Original) The method of Claim 1, wherein the first process fails to establish a connection with the second process because the second process is not running.

5. (Currently Amended) The method of Claim 1, wherein the first rendezvous file being missing from the file system indicates that the second process is not running.

6. (Original) The method of Claim 1, further comprising:

when the first process is started, the first process determining if a fourth process of the first application is running;

the first process requesting the fourth process to perform a task for the first process if the fourth process is running; and

the first process exiting after requesting the fourth process to perform the task for the first process.

7. (Original) The method of Claim 6, wherein the first process determining if the fourth process of the first application is running comprises:

the first process of the first application determining a name of a second file in the file system of the computer, the name of the second file being associated with the first application.

8. (Original) The method of Claim 7, wherein the second file being missing from the file system indicates that the fourth process of the first application is not running.

9. (Original) The method of Claim 8, wherein the second file contains information for the first process to connect to a fourth process for inter-process communication;

failure in connecting to the fourth process using the information contained in the second file indicates that the fourth process of the first application is not running; and

success in connecting to the fourth process using the information contained in the second file indicates that the fourth process of the first application is running.

10. (Original) The method of Claim 1, wherein the first process communicates with the second process using the first connection through an Application Program Interface (API).

11. (Original) The method of Claim 10, wherein the Application Program Interface (API) is platform independent.

12. (Original) The method of Claim 1, further comprising:

when the second process is started, the second process determining if a fourth process of the second application is running;

the second process requesting the fourth process to perform a task for the second process if the fourth process is running; and

the second process exiting after requesting the fourth process to perform the task for the second process.

13. (Currently Amended) An apparatus comprising a single processor on a single computer for inter-process communication between at least two application processes on [[a]] the single processor ~~of a single computer~~, comprising the steps of:

a module for a first process of a first application determining a name of a first rendezvous file in a file system of the computer, the name of the first rendezvous file being associated with a second application, the first rendezvous file containing information for the first process to connect to a second process of the second application for inter-process communication;

a module for the first process initiating a first connection to the second process using the information contained in the first rendezvous file;

a module for the first process initiating communication with the second process using the first connection if the first connection is successfully established;

commanding ~~communicating a task of said first application to said~~ the second application to perform a task for the first application if the first connection is successfully established;

transmitting a set of keys and values between the second application and the first application;

after commanding ~~said~~ the second application to perform ~~said~~ the task, closing ~~said~~ the first application,

wherein multiple instances of an application ~~program~~ process combine into one running instance of ~~said~~ the application ~~program~~ process; and

wherein ~~said~~ the inter-process communication between at least two application processes is restricted to communications between applications running on ~~said~~ the single processor of ~~said~~ the single computer under control of a single instance of an operating system.

14. (Currently Amended) The apparatus of Claim 49, further comprising:

a module for the first process initiating a second connection to the third process using the information in the first rendezvous file, in response to the third process informing the first process that the third process is ready for a connection.

15. (Previously Presented) The apparatus of Claim 49, wherein the third process is started in a server mode without a user interface.

16. (Original) The apparatus of Claim 13, wherein the first process fails to establish a connection with the second process because the second process is not running.

17. (Currently Amended) The apparatus of Claim 13, wherein the first rendezvous file being missing from the file system indicates that the second process is not running.

18. (Original) The apparatus of Claim 13, further comprising:

a module for the first process determining if a fourth process of the first application is running, when, the first process is started;

a module for the first process requesting the fourth process to perform a task for the first process if the fourth process is running; and

a module for the first process exiting after requesting the fourth process to perform the task for the first process.

19. (Original) The apparatus of Claim 18, wherein the module for the first process determining if the fourth process of the first application is running comprises:

a module for the first process of the first application determining a name of a second file in the file system of the computer, the name of the second file being associated with the first application.

20. (Original) The apparatus of Claim 19, wherein the second file being missing from the file system indicates that the fourth process of the first application is not running.

21. (Original) The apparatus of Claim 20, wherein the second file contains information for the first process to connect to a fourth process for inter-process communication; failure in connecting to the fourth process using the information

contained in the second file indicates that the fourth process of the first application is not running; and success in connecting to the fourth process using the information contained in the second file indicates that the fourth process of the first application is running.

22. (Original) The apparatus of Claim 13, wherein the module for the first process communicates with the module for the second process using the first connection through an Application Program Interface (API).

23. (Original) The apparatus of Claim 22, wherein the module for the Application Program Interface (API) is platform independent.

24. (Original) The apparatus of Claim 13, further comprising:

a module for the second process determining if a fourth process of the second application is running, when the second process is started;

a module for the second process requesting the fourth process to perform a task for the second process if the fourth process is running; and

a module for the second process exiting after requesting the fourth process to perform the task for the second process.

25. (Currently Amended) A program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform a method of inter-process communication between at least two application processes on a single processor of a single_computer, the method comprising the steps of:

a first process of a first application determining a name of a first rendezvous file in a file system of the computer, the name of the first rendezvous file being associated with a second application, the first rendezvous file containing information for the first process to connect to a second process of the second application for inter-process communication; and

the first process initiating a first connection to the second process using the information contained in the first rendezvous file;

the first process communicating with the second process using the first connection if the first connection is successfully established; and

the first process starting a third process of the second application if the first process fails to establish a connection with the second process,

wherein ~~said~~ the step of communicating communicates a task of ~~said~~ the second application to ~~said~~ the first application, wherein ~~said~~ the task comprises transmitting a set of keys and values between ~~said~~ the second application and ~~said~~ the first application, wherein ~~said~~ the step of communicating uses strings to transmit ~~said~~ the keys and ~~said~~ the values,

wherein ~~said~~ the step of communicating ~~said~~ the task of ~~said~~ the second application to ~~said~~ the first application further comprises use of a channel,

wherein ~~said~~ the inter-process communication between at least two application processes is restricted to communications between applications running on ~~said~~ the single processor of ~~said~~ the single computer under control of a single instance of an operating system,

wherein the step of communicating the task of the second application to the first application results in one and only one instance of an application runs on the single processor of the computer.

26. (Currently Amended) The medium of Claim 25, wherein the method further comprises the step of:

the first process initiating a second connection to the third process using the information in the first rendezvous file, in response to the third process informing the first process that the third process is ready for a connection.

27. (Original) The medium of Claim 26, wherein the third process is started in a server mode without a user interface.

28. (Original) The medium of Claim 25, wherein the first process fails to establish a connection with the second process because the second process is not running.

29. (Currently Amended) The medium of Claim 25, wherein the first rendezvous file being missing from the file system indicates that the second process is not running.

30. (Original) The medium of Claim 25, wherein the method further comprises the steps of:

when the first process is started, the first process determining if a fourth process of the first application is running;

the first process requesting the fourth process to perform a task for the first process if the fourth process is running; and

the first process exiting after requesting the fourth process to perform the task for the first process.

31. (Original) The medium of Claim 30, wherein the first process determining if the fourth process of the first application is running comprises:

the first process of the first application determining a name of a second file in the file system of the computer, the name of the second file being associated with the first application.

32. (Original) The medium of Claim 31, wherein the second file being missing from the file system indicates that the fourth process of the first application is not running.

33. (Original) The medium of Claim 32, wherein the second file contains information for the first process to connect to a fourth process for inter-process communication;

failure in connecting to the fourth process using the information contained in the second file indicates that the fourth process of the first application is not running; and

success in connecting to the fourth process using the information contained in the second file indicates that the fourth process of the first application is running.

34. (Original) The medium of Claim 25, wherein the first process communicates with the second process using the first connection through an Application Program Interface (API).

35. (Original) The medium of Claim 34, wherein the Application Program Interface (API) is platform independent.

36. (Original) The medium of Claim 25, wherein the method further comprises the steps of:

when the second process is started, the second process determining if a fourth process of the second application is running;

the second process requesting the fourth process to perform a task for the second process if the fourth process is running; and

the second process exiting after requesting the fourth process to perform the task for the second process.

37. (Currently Amended) The method of Claim 1, wherein ~~said~~ the process determines a name of ~~a~~ the first rendezvous file from a filename of ~~said~~ the first application, wherein ~~said~~ the second application locates ~~said~~ the first rendezvous file, without prior knowledge of a running instance of ~~said~~ the first application program, through use of ~~said~~ the filename.

38. (Currently Amended) A method of inter-process communication between at least two application ~~processes~~ programs on a single processor of a single computer, comprising the steps of:

initiating a first application program on said single processor;

upon initiating a second application program, said second application program using a rendezvous file while determining if a prior instance of the second application program is running on said single processor;

if said first application program comprises a prior instance of said second application program and said first program is still running, performing all of:

establishing an inter-process communication channel between said second application program and said first application program;

commanding ~~communicating a task of~~ said ~~second~~ first application program to perform a task for said ~~first~~ second application program by transmitting a set of keys and values between said second application program and said first application program;

after commanding said first application program to perform said task, exiting said second application program,

wherein multiple instances of an application program combine into one running instance of said application program running on said single processor of said single computer.

39. (Cancelled)

40. (Currently Amended) The method of Claim ~~39~~ 38, wherein said rendezvous file comprises a filename named after said first application program, wherein said second application program can locate said rendezvous file, without prior knowledge of a running instance of said first application program, through use of said filename.

41. (Previously Presented) The method of Claim 40, wherein said rendezvous filename comprises placement in a predetermined location, wherein said second application program can locate said rendezvous file, without prior knowledge of a running instance of said first application program, through use of both said filename and said location.

42. (Currently Amended) The method of Claim ~~39~~ 38, further comprising a step of said second application ~~process~~ program computing a filename of said rendezvous file from a name of said second application ~~process~~ program and using said filename in said step of establishing said inter-process communication channel between said second application program and said first application program.

43-44. (Cancelled)

45. (Currently Amended) The method of Claim [[44]] 38, further comprising a step of using strings to transmit said keys and said values.

46. (Previously Presented) The method of Claim 38, wherein said step of communicating said task of said second application program to said first application program comprises use of a channel.

47. (Previously Presented) The method of Claim 46, wherein said channel connects a server process of said second application program to a client process of said first application program.

48. (Previously Presented) The method of Claim 38, wherein said method of inter-process communication comprises communication based upon a filename, wherein a software application based upon said method is not operating system-dependent when said operating system provides access to files.

49. (Previously Presented) The apparatus of Claim 13, further comprising a step of:

a module for the first process starting a third process of the second application if the first process fails to establish a connection with the second process.

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Qing-Yuan Wu whose telephone number is (571)272-3776. The examiner can normally be reached on 8:30am-6:00pm Monday-Thursday and alternate Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

/Qing-Yuan Wu/
Examiner, Art Unit 2194